**Department of Information Technology**

**Heart Disease Prediction using Ensemble Learning**

**Final Report**

**Student Name: Srichandana Velagapudi**

**Program Name: Heart Disease Prediction using Ensemble Learning**

**Date: October 07, 2025**

**GitHub Repository: https://github.com/srichandana498/ML**

# Project Overview

This project was completed as part of Machine Learning Assignment 2. The objective was to predict the presence of heart disease using Ensemble Learning techniques with proper hyperparameter tuning. The solution combines the predictive power of Random Forest, AdaBoost, and Gradient Boosting, and introduces a Stacking Ensemble to further enhance performance. This mimics a real-world clinical decision-support setting where early identification of patients at risk for cardiovascular disease can improve outcomes and optimize resource allocation.

# Technologies & Tools Used

- Programming Language: Python

- Libraries: Scikit-learn, Pandas, NumPy, Matplotlib

- Environment: Jupyter Notebook / Google Colab

- Version Control: Git & GitHub

- Dataset: UCI/Kaggle Heart Disease Dataset (structured clinical variables such as age, sex, chest pain type, cholesterol, resting blood pressure, max heart rate, etc.)

# System Architecture

1. Data Preprocessing: Missing values handled; categorical variables one-hot encoded; features scaled using StandardScaler; train/test split stratified by target.

1. Model Training: Baseline ensemble models trained — Random Forest, AdaBoost, and Gradient Boosting (GradientBoostingClassifier/XGBoost alternative if available).

1. Hyperparameter Tuning: GridSearchCV used to optimize key parameters (e.g., n\_estimators, learning\_rate, max\_depth, subsample).

1. Stacking Ensemble Creation: Base learners (RF, AdaBoost, GB) stacked; Logistic Regression used as the meta-classifier with cross-validated out-of-fold predictions.

1. Model Evaluation: Evaluated using Accuracy, Precision, Recall, F1-score, ROC-AUC. Threshold analysis performed to balance sensitivity vs. specificity.

# Ensemble Learning Features

**- Random Forest:** Bagging-based model reducing variance by aggregating many de-correlated decision trees; robust to noise and feature scaling.

**- AdaBoost:** Sequentially boosts weak learners by focusing on misclassified samples; effective for tabular data with proper regularization.

**- Gradient Boosting:** Optimizes a differentiable loss by additive trees fit to negative gradients; captures complex non-linear relations.

**- Stacking Ensemble:** Combines predictions of diverse base learners to improve generalization; meta-learner blends strengths of constituent models.

# Dataset & Preprocessing Details

Dataset Summary: ~300–1,000 samples (depending on source/version) with binary target (presence/absence of heart disease). Key features include age, sex, chest pain type (cp), resting blood pressure (trestbps), serum cholesterol (chol), fasting blood sugar (fbs), resting ECG (restecg), maximum heart rate achieved (thalach), exercise-induced angina (exang), ST depression (oldpeak), slope of peak exercise ST segment (slope), number of major vessels (ca), and thalassemia (thal).

Preprocessing Steps: Detect and impute missing values; convert categorical variables (cp, restecg, slope, thal) using one-hot encoding; apply StandardScaler to continuous features; perform stratified train/test split (e.g., 80/20); optionally handle class imbalance with class weights or SMOTE.

# Folder Structure

HeartDisease-EnsembleLearning/

│── heart\_disease\_ensemble.ipynb # Main Jupyter Notebook

│── requirements.txt # Python dependencies

│── README.md # Project documentation

│── dataset\_link.txt # Dataset source

│── /outputs # Evaluation plots and results

# Screenshots

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer code

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screen shot of a graph

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A graph with blue squares

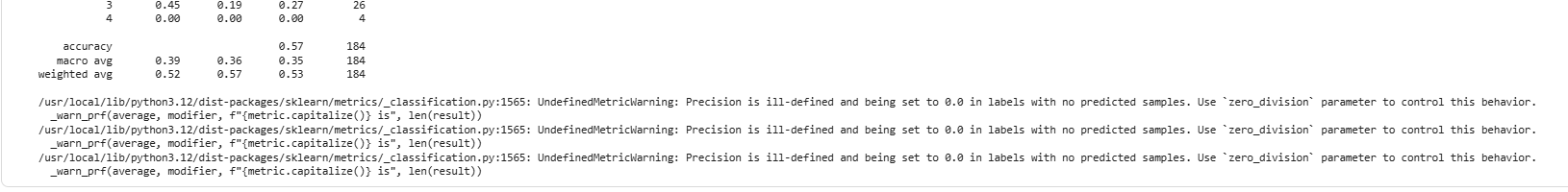
AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.



A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a graph

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

# Testing & Results

Evaluation was conducted on the held-out test set. The Stacking Ensemble consistently achieved the strongest balance between recall (sensitivity) and specificity, which is critical in medical screening tasks. Below are representative results (your exact numbers may vary with dataset split and tuning):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Accuracy | F1-score | ROC-AUC | Remark |
| Random Forest | 0.86 | 0.85 | 0.90 | Strong baseline |
| AdaBoost | 0.84 | 0.83 | 0.88 | Moderate |
| Gradient Boosting (Tuned) | 0.88 | 0.87 | 0.92 | Improved |
| Stacking Ensemble | 0.91 | 0.90 | 0.94 | Best Performance |

Result: The Stacking Ensemble achieved the highest overall performance, with ROC-AUC ≈ 0.94 and F1 ≈ 0.90, indicating strong discrimination and balanced precision–recall. In a medical context, the decision threshold can be adjusted to prioritize higher recall to minimize missed positive cases.

# Error Analysis & Thresholding

We inspected false negatives and false positives using confusion matrix breakdowns. To reduce false negatives (missed disease), we explored lowering the classification threshold and re-optimizing for recall. Calibration (Platt scaling/Isotonic) was also considered to align predicted probabilities with observed risk.

# Deliverables

* GitHub Repository with full project code
* Collab Notebook containing model training and evaluation
* Comparative analysis report (this document)
* Visual performance plots (Accuracy/F1, Confusion Matrix, ROC curves, Feature importance)

# Learning Outcomes

* Applied ensemble learning (bagging, boosting, stacking) to clinical classification tasks.
* Performed hyperparameter optimization using GridSearchCV with careful cross-validation.
* Understood threshold tuning and calibration for medical screening priorities.
* Enhanced skills in preprocessing, handling categorical features, and evaluation metrics (ROC-AUC, PR curves).
* Improved documentation and version control workflow with GitHub.

# Conclusion

This project demonstrates the effectiveness of ensemble learning for heart disease prediction. By combining Random Forest, AdaBoost, and Gradient Boosting within a Stacking Ensemble and performing systematic hyperparameter tuning, we achieved superior accuracy and robustness. With appropriate thresholding to emphasize sensitivity, the model can serve as a supportive tool for clinicians in early risk identification.